

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Malba štětcem v 2D rastrové grafice

Brush Painting in 2D Raster Graphics

Zadání bakalářské práce

Student:

Dalibor Žák

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Malba štětcem v 2D rastrové grafice
Brush Painting in 2D Raster Graphics

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je navrhnout a implementovat techniku malby štětcem s využitím grafického tabletu WACOM v prostředí webu.

Body zadání:

1. Přehled existujících technik malby štětcem.
2. Návrh a implementace vlastního řešení. Předpokládá se implementace více typů štětců s možností uživatelského nastavení.
3. Návrh experimentů a výkonnostních testů.
4. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] M. Pilgrim: HTML5: Up and Running, O'Reilly Media; 1 edition (August 24, 2010), ISBN-13: 978-0596806026
- [2] D. Crockford: JavaScript: The Good Parts, O'Reilly Media; 1st edition (May 2008), ISBN-13: 978-0596517748

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

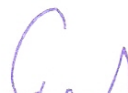
Vedoucí bakalářské práce: **Ing. Petr Gajdoš, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

Dalibor Žák.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 29. dubna 2016

.....*Dalibor Zel*.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Cílem této bakalářské práce je popis webové aplikace pro kreslení ve 2D rastrové grafice určené pro grafické tablety značky Wacom. Součástí práce je popis tabletů Wacom, jejich vlastností a existujících softwarových řešení. Následuje popis jednotlivých součástí aplikace a její popis.

Klíčová slova: Webová aplikace, Wacom, HTML, CSS, Javascript, rastrová grafika, 2D kreslení, grafické tablety

Abstract

Goal of this bachelor thesis is to describe web application for painting in 2D raster graphics intended for graphic tablets produced by Wacom. Part of this thesis is the description of Wacom tablets, their features and existing software. Next, there is description of each component of application and application itself.

Key Words: Web application, Wacom, HTML, CSS, Javascript, raster graphics, 2D painting, graphic tablets

Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	8
Seznam tabulek	9
1 Úvod	11
2 Tablety Wacom	12
3 Existující programy	14
3.1 Bamboo Paper	14
3.2 Adobe Photoshop Elements	15
3.3 Corel Painter Essentials	16
3.4 Krita Desktop	16
3.5 MyPaint	17
3.6 GIMP	18
4 Použité technologie a knihovny	19
4.1 Webový plugin Wacom	19
4.2 HTML	19
4.3 CSS	20
4.4 Javascript	21
4.5 Highcharts	21
4.6 ProColor	23
5 Popis aplikace	24
5.1 Vzhled a ovládání	25
5.2 Štětce	26
5.3 Kreslicí plátno	30
6 Srovnání	32
7 Testování	34
8 Závěr	36
Literatura	37

Seznam použitých zkratk a symbolů

CMYK	– Barevný model využívající k míchání barev čtyři barvy — žlutou, azurovou, purpurovou a černou; využívá se při tisku
HSV	– Barevný model využívající k míchání barevný tón, sytost a jas barvy
LCD	– Obrazovka využívající technologii tekutých krystalů
NPAPI	– Architektura zásuvných modulů využívaná ve webových prohlížečích; v současnosti se od ní upouští
PDF	– Souborový formát pro ukládání dokumentů nezávisle na platformě, kde byly pořízeny
RGB	– Barevný model k míchání barev pomocí tří barev — červené, zelené a modré
USB	– Univerzální sériová sběrnice; nejpoužívanější způsob připojení periférií k počítači

Seznam obrázků

1	Tablet Intuos Pro	12
2	Pero k tabletu Wacom	13
3	Bamboo Paper	15
4	MyPaint štětce	17
5	GIMP — nastavení tabletu	18
6	Nástroj ProColor	23
7	Lišta nástrojů	25
8	Nastavení štětce	26
9	Přítlak pera	30
10	Nový obrázek	30
11	Testovací kresba vytvořena ve webové aplikaci	33
12	Testovací kresba vytvořena v programu MyPaint	33
13	Odezva při kreslení štětcem	35
14	Odezva při změně měřítka	35
15	Odezva při vytváření nového obrázku a při ukládání	35

Seznam tabulek

1	Srovnání modelů Intuos a Intuos Pro	13
2	Proměnné pluginu Wacom	20
3	Vlastnosti grafů Highcharts	22
4	Součet hodnot, při generování štětce	27
5	Odezva při kreslení štětcem	35
6	Odezva při změně měřítka	35
7	Odezva při vytváření nového obrázku a při ukládání	35

Seznam výpisů zdrojového kódu

1	Přidání pluginu do HTML kódu	19
2	Přístup k proměnným u webového pluginu	19
3	Ukázka HTML kódu	20
4	Ukázka CSS kódu	20
5	Deklarace objektu Highcharts	22
6	Vytvoření nástroje pro výběr barev	22
7	Kreslení štětcem	25
8	Náklon pera	28
9	Přítlak pera	29
10	Vytvoření souboru .png	31

1 Úvod

V průběhu 2. poloviny 20. století došlo vedle rozvoje klasického umění k rozvoji digitálního umění. Už v polovině 20. století vznikaly pokusy o generování obrazu počítačem. Největší rozvoj digitální umění zažilo na přelomu 80. a 90. let. Mohl za to rozvoj digitálních technologií a prudký rozvoj grafických programů a komponentů, které se stále zdokonalují. V průběhu let vzniklo mnoho grafických programů, které s postupným rozvojem techniky umožňovaly umělcům stále nové možnosti.[1] Nicméně zpočátku jen těžko mohl nějaký program nahradit plně papír a štětec. Postupný rozvoj grafického hardwaru však umožnil stále více se při kreslení na počítači přibližovat skutečnému kreslení. Toto bylo umožněno nástupem polohovacích zařízení v podobě grafických tabletů. První grafické tablety podobající se těm současným vznikly už na přelomu 50. a 60. let. Nicméně tehdejší systémy byly velmi drahé a náchylné k rušení vnějšími signály. Na počátku 80. let vytvořila společnost Apple první grafický tablet, který byl postupně použitelný prakticky na všechny domácí počítače s grafickou podporou. Postupem času byly tablety zdokonalovány a dostávaly nové funkce, jakými jsou například rozpoznávání rukopisu nebo tlačítka na tabletech.[8]

V této práci se budeme věnovat grafickým tabletům japonské společnosti Wacom, které díky svým možnostem a řadě programů umožňují velmi realistické vytváření obrazů na počítači. Cílem této bakalářské práce bylo navrhnout a vytvořit webovou aplikaci pro kreslení přímo ve webovém prohlížeči s podporou grafických tabletů Wacom. V první části práce se budeme věnovat samotným tabletům Wacom, jejich možnostem a některých modelových řadách. Dnes existuje celá řada grafických programů, ovšem jen některé dokáží plně spolupracovat s grafickými tablety a plně je využívat. Těmto programům a jejich srovnání se budeme věnovat ve druhé části. Třetí část práce bude věnována použitým technologiím při vývoji aplikace. Další část bude věnována popisu samotné aplikace. Bude zde popsán postup vývoje a jednotlivá řešení aplikace. Poslední část se bude věnovat testování jednotlivých částí aplikace a srovnání výsledků s již existujícími programy.

2 Tablety Wacom

Wacom je japonská společnost, založená v roce 1983, specializující se na výrobu grafických tabletů.[13] Většina modelů je prodávána s balíkem programů, které využívají plně výhod a možností tabletů. Tablety jsou prodávány s kompatibilním digitálním perem, které slouží k ovládání. Pero je vyobrazeno na obrázku 2. Některá pera jsou vybavena tlačítky na straně sloužící jako tlačítka myši nebo mají použitelný druhý konec pera, který funguje jako guma. Jednotlivé možnosti a funkce pera se liší dle modelu. Patří mezi ně přitlak, rotace pera. Všechny současné modely používají k propojení s počítačem USB nebo Bluetooth. Firma Wacom v současnosti produkuje několik modelových řad grafických tabletů. Tablety Wacom se řadí mezi takzvané pasivní tablety. To znamená, že využívají technologii elektromagnetické indukce, kde horizontální a vertikální vodiče působí jako vysílací i přijímací cívky. Tablet generuje elektromagnetický signál, který je přijat obvodem v peru. Vodiče v tabletu pak přepnou na přijímací režim a přečtou signál generovaný perem.[2]

Základní modelovou řadu tvoří tablety řady Intuos a Intuos Pro. Obě řady jsou si vzhledově velmi podobné. Základním rozdílem je, že modely Intuos Pro jsou určeny spíše profesionálním uživatelům. Tomu taky odpovídají jejich funkce. Srovnání jednotlivých parametrů nabízí tabulka číslo 1. Dále všechny modely disponují možností přepnutí tabletu pro leváky i praváky. Některé modely navíc disponují vícedotýkovým ovládáním. Obě řady mají k dispozici bezdrátové pero s 0,5mm přesností. Na obrázku 1 je vyobrazen model z řady Intuos Pro. Pro tablety Intuos a Intuos Pro je typický hrubý povrch dotykové plochy, který umožňuje přesnější vedení pera.[11]



Obrázek 1: Tablet Intuos Pro

[9]

Pro účely bakalářské práce byl použit tablet řady Intuos Pro Medium. Tento model disponuje 2048 úrovněmi přitlaku pera, snímáním náklonu pera v 60 úrovních, 8 programovatelnými tlačítky + multifunkčním navigačním prstencem. Aktivní plocha tabletu má rozměry 224 x 140 mm a má rozlišení 2000 řádků na cm. Pero disponuje dvěma tlačítky po straně a má funkční oba konce.

Parametr	Intuos	Intuos Pro
Přítlak pera	1024 úrovní	1024-2048 úrovní
Náklon pera	není	50-60 úrovní
Rozlišení	1000 řádků na cm	až 5080 řádků na cm
Tlačítka	4	6-8 + navigační prstenec
Rozměry aktivní plochy	14,7 x 9,2 cm	závisí na modelu

Tabulka 1: Srovnání modelů Intuos a Intuos Pro

Další rozšířenou modelovou řadou je řada Cintiq. Cintiq je hybridní zařízení v podobě grafického tabletu s vestavěnou LCD obrazovkou, která umožňuje uživateli kreslení přímo na povrch displeje. Tablety jsou nabízené v několika velikostech s různým rozlišením obrazovky a nabízí až 2048 úrovní přítlaku a různou citlivost pera.[11]

Dalším zařízením vyráběné firmou Wacom je Inkling. Inkling umožňuje umělcům vytvářet kresby na papír, které mohou být následně převedeny na digitální obraz. Inkling se skládá z přijímače, který umělec vloží na libovolný papír a speciálního pera, které používá skutečný inkoust. Jakmile skončí s kreslením, tak mohou připojit přijímač do USB portu. Výsledek lze následně exportovat do celé řady obrazových formátů. Ale mohou být také konvertovány pro použití v dalších kreslicích aplikacích, jako například Photoshop nebo Adobe Sketchbook Pro.[11]



Obrázek 2: Pero k tabletu Wacom

[12]

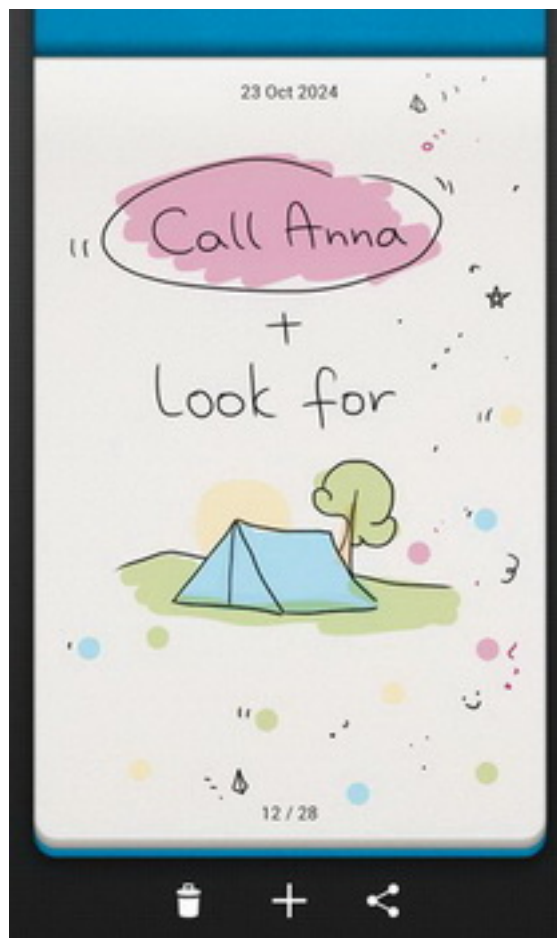
3 Existující programy

Grafické tablety je možné používat pro jejich přesnost prakticky v každé kreslicí aplikaci. Za dobu výroby grafických tabletů vznikla řada aplikací, které umožňují uživateli využívat funkce, kterými tablety disponují. Možnosti a podpora tabletů se liší v závislosti na aplikaci, placené aplikace většinou nabízí uživateli větší spektrum možností. Pro popis v následujících odstavcích bylo vybráno pět desktopových aplikací. Mezi nimi jsou volně dostupné i placené aplikace. Žádná webová aplikace se mezi popisovanými programy nenachází, protože i když existuje řada kreslicích aplikací, neexistuje žádná, která by disponovala podporou grafických tabletů. U každého programu je ve stručnosti uveden popis uživatelského rozhraní, možností a funkcí při kreslení a jak je v programu implementována podpora grafických tabletů.

3.1 Bamboo Paper

Bamboo Paper je jednoduchý bezplatný program pro kreslení z balíku programů Bamboo. Balík programů Bamboo je volně dostupný a obvykle je součástí ovladačů pro grafické tablety Wacom. Program umožňuje uživateli vytvoření dokumentu formou jednoduchého zápisníku za pomoci kreslicích nástrojů. Při spuštění program dostane uživatel k dispozici nový zápisník se vzorovou stránkou. U zápisníku si může zvolit barvu obálky a vzhled stránek, od čistých přes linkované až po notový zápis. Dále může uživatel přidávat nové stránky a kreslit na ně. Ke kreslení lze používat myš i grafické tablety, pro které je program přizpůsoben. Program disponuje třemi jednoduchými nástroji, a to perem, zvýrazňovačem a gumou. Pro použití gumy je třeba zvolit příslušný nástroj na liště, oba konce pera využívají vždy stejný zvolený nástroj. U jednotlivých nástrojů, kromě gumy, je možné zvolit tloušťku ze tří možností a je na výběr pouze z devět přednastavených barev. Volba vlastní barvy mícháním pomocí RGB či HSV zde není k dispozici. Všechny tři nástroje dovedou spolupracovat s tabletem a snímat přítlak pera. Uživatel může navíc importovat do zápisníku obrázky ze souboru, nejvýše může importovat 5 obrázků na každou stránku. Náklon pera program snímat nedovede. Výsledný obrázek má možnost uživatel uložit jako Bamboo Paper (pro pozdější úpravy) nebo jej uložit do formátu PDF, popřípadě jej přímo vytisknout. Na obrázku 3 naleznete snímek z programu.[14]

Součástí balíku programů Bamboo je také nástroj pro rozpoznávání rukopisu. Nástroj je určen především pro zařízení jako jsou grafické tablety. Tento nástroj dokáže rozpoznat psané písmo a převést je do textové podoby. Program má k pro rozpoznávání rukopisu k dispozici několik jazyků včetně českého. Navíc program disponuje virtuální klávesnicí jejíž rozložení odpovídá aktuálně zvolenému jazyku.



Obrázek 3: Bamboo Paper
[10]

3.2 Adobe Photoshop Elements

Adobe Photoshop Elements je rastrový grafický editor určený pro úpravu a vytváření obrázků. Software je nabízen pod shareware¹ licencí. Obsahuje většinu prvků a nástrojů jako profesionální verze, ale s méně funkcemi a jednoduššími možnostmi. Například ve srovnání s klasickým Photoshopem neumožňuje export souborů do barevného modelu CMYK a podporuje zjednodušený systém správy barev. Rovněž obsahuje omezený počet Photoshop pluginů. Program je přizpůsobený používání grafických tabletů. Umožňuje detailní nastavení a přizpůsobení funkcí tabletu a pera. Uživatel takto může nastavit citlivost pera a snímání přitlaku a funkci jednotlivých tlačítek na tabletu i peru. Dále umožňuje uživateli nastavení jednotlivých vlastností štětců pro co nejrealističtější simulaci. Jedná se o nastavení hustoty štětín, blednutí barvy, tvrdost, rozptyl, úhel a zaoblení.[3]

¹Software s bezplatnou zkušební dobou

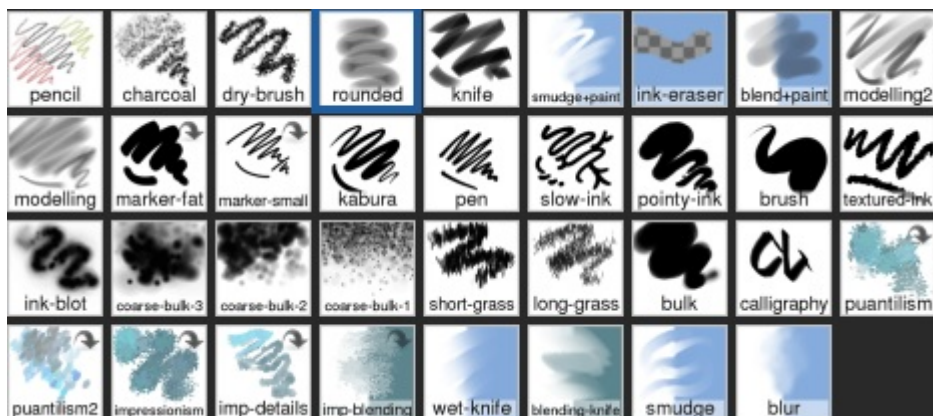
3.3 Corel Painter Essentials

Corel Painter Essentials je softwarové studio pro kreslení a přeměnu fotografií v malby pro domácí použití. Program nabízí uživateli 2 pracovní plochy. Jednu pro běžné kreslení, druhou určenou pro přeměnu fotografie v jeden z typů nabízené malby. Uživatel má k dispozici několik různých kategorií štětců. Uživatel má také na výběr nedávno použité štětce. K dispozici je i několik různých stylů maleb, například olejomalbu, vodové barvy spolu s tužkou, pastelem, křídou a řadou dalších. Program rovněž nabízí výběr z několika podkladů pro kreslicí plátno, výběr barvy pomocí HSV či RGB míchání a rotaci kreslicího plátna kdykoliv během kreslení, čímž se program snaží nabídnout uživateli stejné možnosti, jako kdyby uživatel používal skutečný papír a kreslicí techniky. Program je od verze 4 plně kompatibilní se všemi produkty Wacom. Z tabletu Wacom po konfiguraci dokáže snímat přítlak i náklon pera a tím plně přizpůsobit rozložení štětin ve štětcu. Z tabletů také dokáže snímat vícedotyková gesta pro manipulaci s plátnem.[15]

3.4 Krita Desktop

Krita Desktop je open source¹ kreslicí aplikace. Aplikace je dostupná ve verzích pro operační systémy Windows, Linux i Mac OS. I přesto, že se jedná o volně dostupný program, tak nabízí uživatelské rozhraní s celou řadou nástrojů a funkcí. Při vytváření nového obrázku má uživatel možnost volby rozměrů, barvy pozadí a navíc průhlednosti pozadí. Průhlednost lze rovněž aplikovat i při samotném kreslení. Po levé straně má k dispozici uspořádány důležité nástroje. Vedle kreslení štětcem, je zde možnost práce s textem v obrázku. Kreslení pomocí kaligrafického štětce nebo práce se základními tvary. Další panel nástrojů se objeví po kliknutí pravým tlačítkem. Tento panel obsahuje několik typů psacích nástrojů. Mezi nimi se nachází například klasický štětec, guma, zvýrazňovač, pero a tužka. Vedle toho má zde uživatel dva sloty pro barvu, mezi nimiž lze přepínat. Každý slot má potom k dispozici nastavení barvy popředí a pozadí. Program pracuje s vrstvami, které lze libovolně přidávat a odebírat. Jednotlivé nakreslené objekty jsou neustále kdykoliv přístupné a lze je upravovat nebo mazat. Vedle toho je zde uživateli nabídnuto rozsáhlé možnosti v nastavení programu, které umožňují detailně nastavit chování štětce. Nevýhodou tohoto programu je dlouhá odezva na slabých počítačích a velmi dlouhá inicializace při spouštění ve srovnání s jinými podobnými programy.[16]

¹ Aplikace s volně dostupným zdrojovým kódem

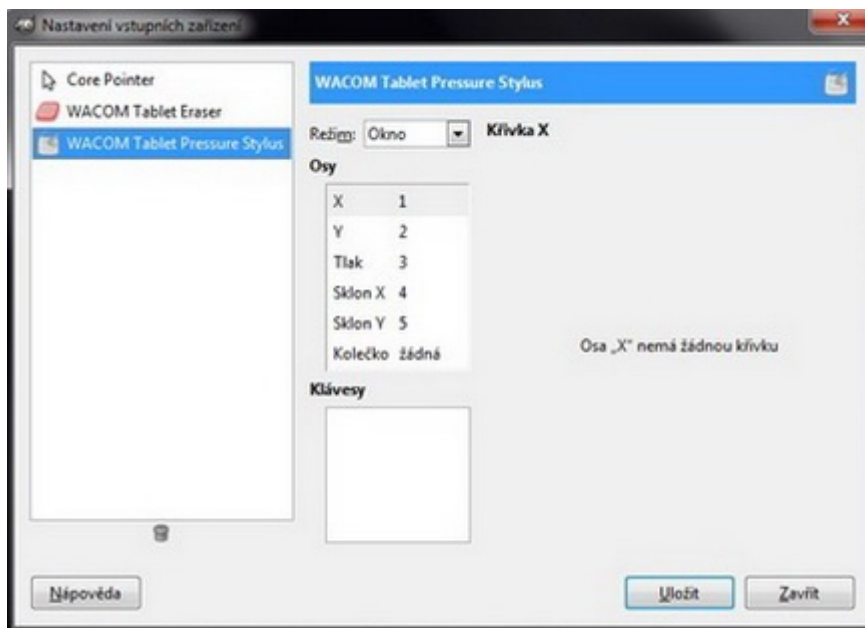


Obrázek 4: MyPaint štětce

[18]

3.5 MyPaint

MyPaint je open source aplikace určená především pro kreslení s grafickými tablety Wacom. Aplikace je stejně jako Krita dostupná pro operační systémy Windows, Linux i Mac OS. Program dobře spolupracuje s tablety Wacom. Dokáže snímat přítlak i náklon pera, nedokáže však snímat druhý konec pera. Pro použití gumy je potřeba výběr nástroje. Jednotlivé nástroje jsou uspořádány přehledně a snadno dostupné a ovladatelné pomocí pera. Uživatel má k dispozici řadu kreslicích nástrojů, které věrně fyzikálními vlastnostmi napodobují skutečnou předlohu. Na obrázku 4 je ukázka všech druhů štětců, které aplikace nabízí. Uživatel má možnost obrázek přibližovat i oddalovat, posouvat s ním po ploše obrazovky a rotovat s ním. Ovládání těchto nástrojů je řešeno, tak aby je bylo možno používat za pomoci ovládacího pera tabletu. Nechybí práce s vrstvami ani paměť úprav. Navíc je možno program přepnout do režimu celé obrazovky a kreslit po celé ploše monitoru. Program má velmi dobrou odezvu při kreslení, nedochází při rychlém pohybu k přerušení stopy štětce, dobře pracuje s průhledností, chybí zde však možnost nastavení velikosti obrázku při výběru nového.[17]



Obrázek 5: GIMP — nastavení tabletu
[19]

3.6 GIMP

GIMP je volně dostupná multiplatformní aplikace pro úpravu a vytváření rastrové grafiky. Používá se zejména pro tvorbu webové grafiky, úpravu fotografií a podobné účely. Kromě široké škály rastrových nástrojů obsahuje i některé vektorové funkce. GIMP je v současnosti dostupný zdarma včetně zdrojových kódů. GIMP disponuje širokou škálou nástrojů, od štětce přes rozprašovač po gumu, razítko a další. Nabízí práci s kanály a vrstvami. K aplikaci existuje řada zásuvných modulů napsaných v jazyce Python. Plně podporuje většinu standardních rastrových formátů. Umožňuje importovat vektorové obrázky ve formátu SVG nebo dokumenty PDF. GIMP rovněž nabízí uživateli podporu grafických tabletů. Ovšem oproti dříve zmiňovaným programům ovšem neumí rozpoznat připojený tablet automaticky. Program sice dokáže využívat tablet pro kreslení ovšem nelze bez aktivace tabletu v GIMPu jako vstupního zařazení rozpoznat přítlak a náklon pera. Naproti tomu po aktivaci lze plně pracovat oproti většině aplikací i se snímáním náklonu pera. Na obrázku 5 je ukázka nastavení tabletu v GIMPu. Jak je z obrázku patrné, tak je možné provést zvlášť nastavení pro oba konce pera. U každého konce lze nastavit chování jednotlivých os. Nastavování přítlaku nebo náklonu probíhá pomocí funkce, kdy si v grafu úpravou funkce nastavíme, jak se oba parametry projeví při kreslení. Toto je možné provést pro oba konce zvlášť, takže se každý konec pera může chovat za stejných podmínek rozdílně.[4]

4 Použité technologie a knihovny

4.1 Webový plugin Wacom

Wacom nabízí vývojářům velké možnosti v podobě zdarma dostupných knihoven a pluginů nejen pro webové aplikace, ale rovněž nabízí knihovny pro nejpoužívanější operační systémy. Vedle nich jsou k dispozici ukázkové aplikace a rozsáhlá dokumentace.[20] Webový plugin Wacom je určen vývojářům webových aplikací. Umožňuje získat všechna potřebná data a proměnné z tabletu. Přístup k jednotlivým proměnným závisí na modelu tabletu a podporovaných funkcích. Pro přístup k datům je nutné použít plugin jako součást javascriptové aplikace. Poté může být použit s jakoukoliv komponentou podporovanou webovým prohlížečem jakou je například Canvas, ale lze jej rovněž použít spolu s dalšími pluginy (například Adobe Flash) nebo frameworky (například Silverlight). Plugin je možné použít spolu s jakýmkoliv prohlížečem, který podporuje NPAPI architekturu, popřípadě ActiveX architekturu v prohlížečích Internet Explorer.[21]

Pokud chceme použít plugin musíme nejdříve deklarovat objekt v HTML souboru, viz výpis 1. Dále je nutné zkontrolovat, zda je plugin načten.

Pokud chceme získat či zapsat data do proměnných, můžeme to udělat podle výpisu 2. Kód ukazuje jak získat hodnotu tlaku pera a jeho souřadnice X a Y.

Celkem plugin nabízí přístup až k 19 různým proměnným v závislosti na modelu tabletu. Seznam a popis těch nejdůležitějších je uveden v tabulce číslo 2.

```
<object id="wtPlugin" type="application/x-wacomtabletplugin">
  <param name="onload" value="pluginLoaded">
</object>
```

Výpis 1: Přidání pluginu do HTML kódu

```
var plugin = document.getElementById('wtPlugin');
var pressure = plugin.penAPI.pressure;
var positionX = plugin.penAPI.posX;
var positionY = plugin.penAPI.posY;
```

Výpis 2: Přístup k proměnným u webového pluginu

4.2 HTML

HTML je v informatice název značkovacího jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. HTML je hlavním jazykem pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu

Jazyk HTML je charakterizován množinou značek (tzv. tagů) a jejich vlastností (atributů). Značky jsou obvyklé párové přičemž koncová značka je shodná s počáteční, jen s lomítkem před názvem. Ve výpisu 3 je uveden příklad pro zápis odstavce.[5]

Název	Datový typ	Popis
pointerType	long	Typ ukazatele (pero, myš, guma)
posX	long	Poslední známá souřadnice X
posY	long	Poslední známá souřadnice Y
pressure	long	Přítlak pera v rozsahu 0,0 až 1,0
tiltX	float	Náklon pera v ose X v rozsahu -1,0 až 1,0
tiltY	float	Náklon pera v ose Y v rozsahu -1,0 až 1,0
rotationDeg	float	Rotace pera ve stupních
rotationRad	float	Rotace pera v radiánech
version	long	Verze pluginu

Tabulka 2: Proměnné pluginu Wacom

<p>Text odstavce</p>

Výpis 3: Ukázka HTML kódu

V současnosti je k dispozici verze HTML 5, která přináší vývojářům webových stránek nové možnosti. V mnohých případech jde o jednodušší zápis kódu a především řada nových tagů a atributů.[6] Příkladem je tag *canvas*, který slouží k vytvoření kreslicího plátna pro dynamické vykreslování 2D grafiky. Jedná se o hlavní součást webové stránky mé aplikace. Tento tag umožňuje velmi jednoduchou práci se 2D grafikou a tím je ideální pro podobné kreslicí aplikace. Součástí tagu *canvas* jsou dva parametry *height* a **width**, které slouží pro nastavení rozměru plátna. Další důležitou součástí jsou události *onmouseup*, *onmousemove* *onmousedown*, které umožňují rozeznat stisknutí nebo uvolnění tlačítka myši uvnitř plátna, popřípadě táhnutí myši.

4.3 CSS

CSS je v informatice jazyk navržený pro popis způsobu zobrazení elementů na webových stránkách. Hlavním smyslem jazyka je oddělit vzhled dokumentu od jeho struktury a obsahu. Definice kaskádových stylů se sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarací, jednotlivé deklarace uvnitř bloku jsou odděleny středníky. Příklad pro zápis pravidla v CSS je uveden ve výpisu 4.[6]

```
body {
  background-color: white;
  color: black;
  padding: 10px !important;
}
```

Výpis 4: Ukázka CSS kódu

V současnosti je vývojářům k dispozici verze CSS 3, která nabízí zcela nové vlastnosti a možnosti. Jedná se o stínování prvků a textů, zaoblení rohů a transformace (pohyb a změna velikosti prvků).

4.4 Javascript

Javascript je multiplatformní scriptovací jazyk nyní používaný zpravidla jako interpretovaný programovací jazyk pro WWW stránky. Slouží k vytváření interaktivních prvků uživatelského rozhraní, animace nebo efekty objektů. Syntaxí se podobá jazykům C, C++, Java. Slovo Java je však součástí názvu pouze z marketingových důvodů. S programovacím jazykem Java nemá, s výjimkou podobné syntaxe, nic společného. Javascript je klientský script. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonán. Javascript je hlavní součástí vyvíjené aplikace, bez něj by vytvoření aplikace nebylo možné.[7]

Základní vlastností jazyka jsou:

- interpretovaný - nemusí se kompilovat
- objektový - využívá objektů prohlížeče a zabudovaných objektů
- závislý na prohlížeči
- záleží na velikosti písmen v zápisu
- dynamické přiřazování datových typů proměnných
- beztrždní - nemá třídy narušující od jiných objektově orientovaných jazyků

Javascript rovněž nabízí řadu knihoven, oficiálních nebo vytvořených uživateli, které přidávají další možnosti, k tvorbě webových aplikací.

4.5 Highcharts

Highcharts je knihovna napsaná v javascriptu umožňující přidání interaktivních grafů na webovou stránku nebo aplikaci. Uživatel má možnost si vybrat z velkého množství typů grafů. Knihovna je volně dostupná pro nekomerční použití. Je kompatibilní se všemi moderními desktopovými i mobilními webovými prohlížeči. Klíčovou vlastností je, že nezávisle na licenci je uživateli umožněno stáhnout zdrojový kód a provádět vlastní editace. Díky tomu, že je knihovna psaná v čistém javascriptu nevyžaduje další klientské pluginy, které by mohli způsobovat problémy s funkčností. Ke zprovoznění jsou třeba dva soubory, vedle samotné knihovny Highcharts je vyžadována knihovna jQuery (nabízí rozšířené funkce jako práce s DOM elementy, události, manipulace s CSS, efekty a animace).[23]

Konfigurace grafu je velmi jednoduchá. Provádí se vytvořením nového objektu, viz výpis číslo 5. Vytvoříme si proměnnou, zde nazvanou *chart* do níž vytvoříme nový objekt. Uvnitř závorek následně můžeme nastavovat jednotlivé vlastnosti a funkce grafu. Navíc je možné přidat tímto způsobem výchozí data. Ke každé části grafu má k dispozici své vlastní atributy, které dávají uživatelům flexibilitu v možnostech nastavení vzhledu a vlastností grafu. Vedle vzhledu je možné nastavit jednotlivým částem události včetně kódu, který se po vyvolání události provede. V tabulce číslo 3 je přehled hlavních atributů objektu.

chart	Nastavuje vlastnosti celého elementu: místo, kde se zobrazí; rozměry; typ grafu
title	Formátování názvu
xAxis	Formátování vodorovné osy
yAxis	Formátování svislé osy
plotOptions	Zde se nastavují především jednotlivé události (kliknutí nebo táhnutí myši)
series	Nastavení dat a manipulace s nimi

Tabulka 3: Vlastnosti grafů Highcharts

V aplikaci byla knihovna použita pro vytvoření interaktivních grafů, které slouží k nastavování štětců (podrobný popis v odstavci 6).

```

var chart = new Highcharts.Chart({
  chart: {
    renderTo: "xPlot",
    defaultSeriesType: 'spline',
    events: {
      click: function(e) {
        //Kliknutí do prostoru grafu
      }
    },
    width: 150,
    height: 150
  },
  title: {
    //Nastavení názvu grafu
  },
  ...
});

```

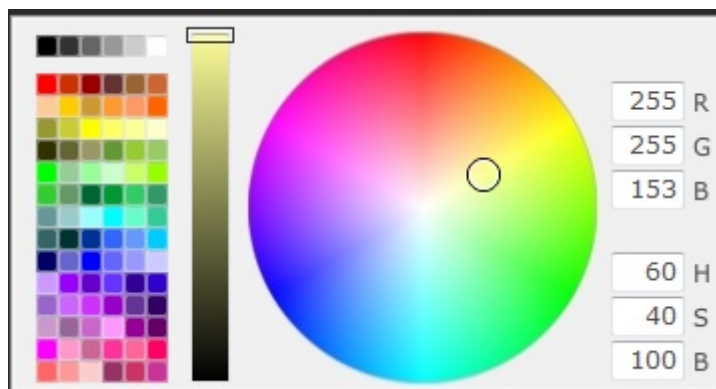
Výpis 5: Deklarace objektu Highcharts

```

ProColor.prototype.attachButton('bgPicker', {
  imagePath:'img/procolor_win_',
  showInField: true,
  color: '#FFFFFF',
  onChanged:function(pc) {
    _background = pc.color;
    _backgroundR = parseInt(_background.substring(1,3), 16);
    _backgroundG = parseInt(_background.substring(3,5), 16);
    _backgroundB = parseInt(_background.substring(5,7), 16);;
  }
});

```

Výpis 6: Vytvoření nástroje pro výběr barev



Obrázek 6: Nástroj ProColor

4.6 ProColor

ProColor je jednoduchá flexibilní javascriptová knihovna, která umožňuje přidat do stránky nástroj pro výběr barev. Nástroj je vyobrazen na obrázku číslo 6. Knihovna je kompatibilní se všemi moderními prohlížeči. Použití knihovny je velmi jednoduché. Inicializace se provádí dvěma způsoby. Zde záleží, zda chceme mít nástroj umístěný přímo na stránce uvnitř nějakého elementu nebo jej chceme provázat s tlačítkem. V prvním případě se inicializace provádí stejným způsobem jako u knihovny Highcharts (viz odstavec 4.5). V opačném případě je nutné uvést, ke kterému tlačítku se nástroj připojí, viz výpis číslo 6. Připojení k tlačítku se provádí metodou `attachButton`, kde jako první parametr je uvedeno id tlačítka a jako druhý parametr objekt s atributy. Ve výpisu 6 je ukázka vytvoření nástroje pro výběr barvy pozadí. Důležitými parametry zde jsou parametr *color*, který slouží k nastavení výchozí barvy a poté parametr *onChanged*, který obsahuje posloupnost příkazů, které se provedou po výběru barvy.[22]

5 Popis aplikace

Při vývoji aplikace byl kladen největší důraz na jednoduché a přehledné ovládání a zároveň, aby aplikace nabídla uživateli co nejvíce možností. Pro tvorbu kreslicí aplikace bylo však zapotřebí nejprve rozhodnout jakým stylem bude obrázek reprezentován v paměti. Pro záznam obrázku v paměti existují v zásadě dvě možnosti, a to vektorový nebo rastrový formát. Vektorová grafika spočívá v tom, že obrázek je rozdělen na základní přesně definované útvary, jako jsou body, přímky a křivky. Mezi výhody vektorové grafiky patří především možnost libovolného zmenšování nebo zvětšování obrázku beze ztráty kvality. Dále je možné s každým objektem pracovat odděleně a výsledná paměťová náročnost je u jednotlivých objektů menší než při použití rastrového zápisu (například černé kolečko se zapíše jako kruh o daném poloměru vyplněný černou barvou - máme tedy 3 informace). Nevýhodou může být, že pokud obrázek překročí určitou mez složitosti, může být vektorová grafika náročnější na procesor a operační paměť. Nehodí se pro zápis složitějších barevných ploch například fotografií. Vektorovou grafiku využívají například formáty *.svg* nebo *.pdf*. Oproti tomu rastrová grafika je typická tím, že celý obrázek je popsán pomocí jednotlivých barevných bodů (pixelů). Body jsou uspořádány do mřížky, kde každý bod má přiřazenou svou polohu a barvu v nějakém barveném modelu (například RGB). Kvalitu záznamu ovlivňuje rozlišení a barevná hloubka¹. Velkou nevýhodou oproti vektorové grafice jsou vysoké nároky na paměť. Při vysokém rozlišení a barevné hloubce může velikost obrázku dosahovat i jednotek megabytů. Například výše zmíněné černé kolečko musí být v paměti zaznamenáno pixel po pixelu, kde každý pixel bude vedle pozice obsahovat totožnou informaci o barvě. Typickým příkladem rastrové grafiky je formát *.bmp*. Velikost obrazu se dá při ukládání částečně snížit kompresí obrazu, kdy jsou odstraněny nedůležité informace. Příkladem komprimovaného obrazového formátu je *.jpg*. Další nevýhodou je zmenšování nebo zvětšování obrázku, které vede ke ztrátě kvality. Výhodou však je, že snadné pořízení obrázku za pomoci skeneru nebo fotografií.

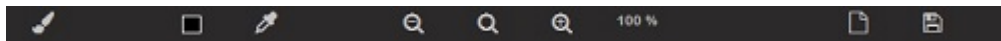
Pro účely aplikace byl zvolen rastrový formát. Jak již bylo zmíněno výše základní součástí HTML stránky je prvek *canvas*, který umožňuje snadné vykreslování 2D grafiky ve webovém prohlížeči. Díky svým funkcím a možnostem je použití canvasu ideální s rastrovou grafikou. Snímáním pozice kursoru uvnitř canvasu se dá velmi snadno přistoupit do paměti k požadovanému pixelu a provést potřebné operace. Navíc vzhledem k implementovaným nástrojům je použití rastrové grafiky ideální. Pro snímání pohybu myši uvnitř canvasu jsou k dispozici 3 události, se kterými lze pracovat. První událostí je *onmousedown*. Tato událost je vyvolána ve chvíli, kdy uživatel klikne do plochy canvasu. Tato událost je pak následně propojena s další událostí — *onmousemove*. Událost *onmousemove* je aktivní pouze v případě, že byla předem vyvolána událost *onmousedown*. Tato událost snímá souvislý pohyb kursoru uvnitř canvasu. Třetí událostí je *onmouseup*, která rozpozná uvolnění tlačítka myši. Ve chvíli, kdy je tato událost v aplikaci vyvolána dojde k nahrazení metody v události *onmousemove* hodnotou *null*. Tím pádem po

¹Počet bitů na zaznamenání barvy jednoho pixelu

uvolnění tlačítka myši nebude událost `onmousemove` již aktivní a canvas nebude reagovat na pohyb myši bez stisknutí tlačítka.

5.1 Vzhled a ovládání

Při návrhu vzhledu aplikace a ovládacích prvků bylo cíleno na uživatele používající grafické tablety. Hlavní předností aplikace je jednoduché a přehledné rozmístění ovládacích prvků, tak aby byly snadno dostupné a ovladatelné za pomoci pera grafického tabletu. Lišta s nástroji je umístěna vždy v horní části stránky a není ovlivněna pohybem po stránce pomocí posuvníků. Navíc relativní velikost lišty vždy zůstává stejná bez ohledu na rozlišení obrazovky. Nejmenším použitelným rozlišením je 800x600. Na nižší rozlišení aplikace není stavěna. Vzhledem k tomu, že tablety Wacom jsou určeny především pro stolní počítače, tak menší rozlišení není zapotřebí. Pod lištou se pak nachází aktivní plocha stránky. Kreslicí plátno je v závislosti na zvolené velikosti vždy umístěno v prostřední části obrazovky. Lišta s nástroji je vyobrazena na obrázku číslo 7. Jak je z obrázku patrné, tak lišta obsahuje celkem 8 tlačítek. Tlačítko zcela vlevo slouží k nastavení tvaru a paramterů štětce. Další dvě tlačítka slouží k nastavování barev. První tlačítko otevře paleta a nástroj pro míchání barev a navíc obsahuje i ukazatel aktuálně zvolené barvy popředí, druhé tlačítko slouží jako kapátko. Další trojice tlačítek slouží ke zmenšování a zvětšování obrázku, prostřední tlačítko vykreslí obrázek v originální velikosti. Nechybí zde ani ukazatel aktuální hodnoty přiblížení. Obrázek lze zvětšit až na 500 procent. Poslední dvě tlačítka slouží pro práci s obrázkem. První tlačítko slouží k vytvoření nového obrázku. Druhé tlačítko vygeneruje z aktivního pole canvasu obrázek ve formátu `.png`, který může uživatel následně uložit na disk.



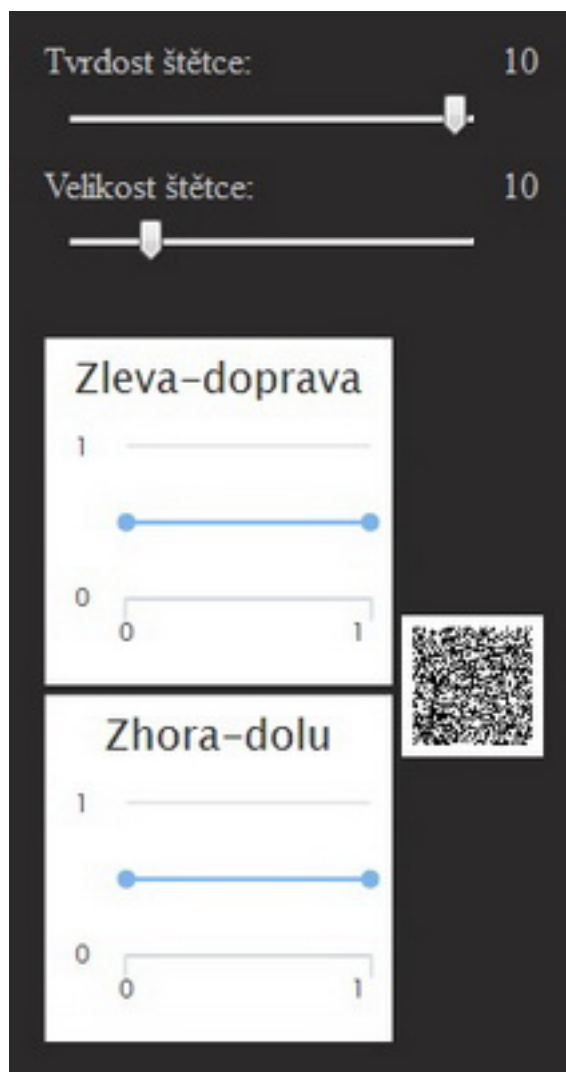
Obrázek 7: Lišta nástrojů

```
ctx. fillStyle = _foreground;
for (i = 0; i < bristleCount; i++) {
    if (Math.floor((y+BristleArrPt[i][1]-_canvasUpCn)/_zoomCurrent) >= 0 &&
        Math.floor((y+BristleArrPt[i][1]-_canvasUpCn)/_zoomCurrent) < _canvasHeight-1 &&
        Math.floor((x+BristleArrPt[i][0]-_canvasLeftCn)/_zoomCurrent) >= 0 &&
        Math.floor((x+BristleArrPt[i][0]-_canvasLeftCn)/_zoomCurrent) < _canvasWidth-1) {

        ctx. fillRect ((x+BristleArrPt[i][0]) - ((x+BristleArrPt[i][0]) \% _zoomCurrent),
            (y+BristleArrPt[i][1]) - ((y+BristleArrPt[i][1]) \% _zoomCurrent),
            _zoomCurrent, _zoomCurrent);

        BrushLayer[Math.floor((y+BristleArrPt[i][1]-_canvasUpCn)/_zoomCurrent)]
            [Math.floor((x+BristleArrPt[i][0]-_canvasLeftCn)/_zoomCurrent)] = _foreground;
    }
}
```

Výpis 7: Kreslení štětcem



Obrázek 8: Nastavení štětce

5.2 Štětce

Jak již bylo zmíněno v předchozím odstavci, první tlačítko slouží k nastavení tvaru a parametrů štětce. Kliknutím na tlačítko se nám otevře nabídka s ovládacími prvky, viz obrázek číslo 8. Základními prvky pro nastavení štětců jsou dva grafy, které slouží k nastavení pozic jednotlivých štětín v horizontálním a vertikálním směru. Vedle kterých je vyobrazena ukázka aktuálně používaného štětce. Každý z těchto grafů obsahuje dvě osy v rozsahu 0 až 1. Vodorovná osa představuje pozici štětiny. V případě vodorovného směru, hodnota 0 označuje levou stranu štětce a v případě svislého směru pak horní stranu štětce. Svislá osa pak označuje pravděpodobnost s jakou se štětina v daném místě vygeneruje. Ve výchozím stavu (na obrázku 8) je vygenerován čtvercový štětec, kdy jednotlivé štětiny jsou vygenerovány s 50% pravděpodobností. Pokud chceme vygenerovat štětiny se 100% pravděpodobností musíme přesunout body tak aby se ve

	x0	x1	x2
y0	x0+y0	x1+y0	x2+y0
y1	x0+y1	x1+y1	x2+y1
y2	x0+y2	x1+y2	x2+y2

Tabulka 4: Součet hodnot, při generování štětce

svislé ose grafu nacházely v hodnotě 1. U obou grafů je možné jednotlivé body funkce posouvat po ploše grafu. Kliknutím na volnou plochu grafu se přidá nový bod a kliknutím na existující bod jej smažeme. Tyto dva grafy jsou dále doplněny o dva posuvníky nastavující tvrdost a velikost štětce. Velikost štětce se dá nastavit v rozsahu od 1 do 50 pixelů. Tvrdost štětce se nastavuje v rozsahu 1 až 10.

Jak již bylo zmíněno generování štětín probíhá na základě nastavení obou grafů. Při vytváření štětce je použita mřížka o rozměrech 51x51, štětec takto může mít až 2601 štětín. Při generování štětců dojde postupně k odečtení hodnot jednotlivých bodů grafu. Následně program dopočítá hodnoty mezi jednotlivými body, tak aby z každého grafu dostal požadovaných 51 hodnot. Základem jsou dvě smyčky *for*. První smyčka se stará o přečtení hodnot existujících bodů grafu. Uvnitř této smyčky se nachází druhá, která pokud je potřeba dopočítá hodnoty mezi dvěma body. Rozestup mezi jednotlivými body závisí na počtu štětín, v tomto případě je to 0,02. Hodnoty získané z obou grafů se následně sečtou, tak že získáme tabulku hodnot o rozměrech odpovídající počtu štětín. Postup při součtu je znázorněn na mřížce 3x3 v tabulce 4, kde písmenem x jsou označeny hodnoty získané z vodorovné osy a písmenem y ze svislé osy. Tyto hodnoty jsou následně normalizovány, takže získáme hodnoty v rozsahu 0 až 1. Díky tomuto hodnoty v podstatě odpovídají pravděpodobnosti s jakou se štětina v daném místě vygeneruje. Hodnota 0 odpovídá 0 % a hodnota 1 pak odpovídá 100 %. Před generováním štětín jsou z tabulky eliminovány hodnoty menší než 0,4. Tímto dojde k odstranění těch nejmenších hodnot. Pokud by toto nebylo použito nebylo by možné vygenerovat jiné tvary štětců než čtvercové. Protože by stále mohlo docházet ke generování štětín po celé ploše. Následně dojde k průchodu tabulkou a u každé hodnoty větší než 0,4 dojde k vygenerování náhodného čísla v rozsahu 0 až 1. Pokud je vytvořené náhodné číslo menší než hodnota v tabulce, tak poté dojde k vytvoření štětiny v daném místě. Hodnoty jsou následně uloženy do paměti a přepočítány tak, aby střed štětce ležel v místě kursoru a navíc jsou hodnoty přenásobeny zvolenou velikostí štětce. Takto získáme štětec požadovaného tvaru a velikosti. Tvar a velikost štětce jsou pak ještě ovlivněny hodnotami získanými z tabletu Wacom. Pokud obrázek přiblížíme, tak velikost štětce se nemění, díky tomu je možné dokreslit do obrázku detaily. Ve výpisu 7 je zobrazen postup při vykreslení stopy štětce. První řádek nastaví barvu, kterou se bude kreslit. Následuje smyčka *for*, která vykreslí stopu všech štětín ve štětcí. Podmínka *if* zajišťuje, aby se kreslilo pouze uvnitř aktivní plochy canvasu. Další dva příkazy potom vykreslí samotnou stopu štětiny. První vykreslí vizuální podobu na obrazovce, druhý uloží do paměti informaci o barvě.

```

var tmpY = Math.abs(tiltY)/3+1;
for (i=0; i < bristleCount; i++) {
    if (tiltY < 0) {
        if (BristleArr [i][1]<0) {
            BristleArrPt [i][1]= BristleArr [i][1]* tmpY;
        }
        else {
            BristleArrPt [i][1]= BristleArr [i][1]*(1/ tmpY);
        }
    }
    else {
        if (BristleArr [i][1]<0) {
            BristleArrPt [i][1]= BristleArr [i][1]*(1/ tmpY);
        }
        else {
            BristleArrPt [i][1]= BristleArr [i][1]* tmpY;
        }
    }
}

```

Výpis 8: Náklon pera

5.2.1 Propjení štětce s tabletem

Při kreslení štětcem dochází k ovlivnění pozic jednotlivých štětin třemi proměnnými získanými z tabletu. Jedná se o přítlak pera, náklon pera ve svislé ose a náklon v horizontální ose. Pro tento účel jsou v paměti vyhrazena dvě pole proměnných, ve kterých jsou uloženy pozice štětin. První pole obsahuje původní rozmístění před aplikací proměnných získaných z pera. V druhém jsou pak uloženy hodnoty ovlivněné perem tabletu.

Jakmile dojde k použití pera uvnitř kreslicího plátna, tak nejprve jsou získány z pera hodnoty náklonu. Náklon pera v obou osách je reprezentován hodnotami -1 až 1. Kladné či záporné znaménko označuje směr náklonu pera v každé ose. Tyto hodnoty jsou následně přepočítány do rozsahu 1 až 3. Tímto se bude snadněji počítat nová pozice štětiny. Vzhledem k tomu, že pro výpočet se používá násobení, tak hodnota 1 bude značit, že štětina zůstane na stejné pozici a hodnota 3 posun na trojnásobek v příslušné ose. Pokud například nakloníme pero vlevo, tak se štětina rozdělí podél svislé osy na polovinu. Štětiny nacházející se na levé polovině se přiblíží blíže k ose, zatímco štětiny vpravo se od osy vzdálí. Pokud nakloníme pero vpravo, tak postup bude opačný. Analogicky to funguje při náklonu pera dopředu nebo dozadu. Výpis 8 ukazuje jak je řešen náklon pera v jedné z os.

Jakmile program získá pozice štětin ovlivněných náklonem, tak dojde k přepočítání pozic podle přítlaku pera a nastavené tvrdosti štětce. Pokud používáme skutečný štětec a papír, tak tvrdost štětce ovlivňuje, jak se štětina ohnou při přitlačení štětce na papír. V aplikaci jak již bylo

zmíněno je možné tvrdost štětce nastavit v hodnotách 1 až 10, kdy nejtvrdší štětec má hodnotu 10 a nejměkčí 1. Pokud máme nastavenou tvrdost štětce 10, tak štětiny nebudou ovlivněny přitlakem vůbec. Čím více hodnotu snížíme, tak tím více se při stejné hodnotě přitlaku štětiny vzdálí od středu štětce. Na obrázku 9 je znázorněna ukázka jak je štětec ovlivněn tlakem pera a nastavenou tvrdostí. Pro ukázkou byl použit štětec velikosti 10. Zhora dolů je nastavena tvrdost 10, 7, 4 a 1. Přítlak se zvyšuje zleva doprava. Ve výpisu 9 je ukázka výpočtu přitlaku pera. Do proměnné tmp je uložena konstanta v rozsahu 1 až 2, kterou bude upravena pozice štětín. Při nastavení tvrdosti na 1 a maximálním přitlaku může tedy dojít až k dvounásobnému zvětšení plochy štětce.

```

if (_brushHardness==10)
    var tmp = 1;
else
    var tmp = 1+(pres*((1/_brushHardness)*2));

for (i=0; i < bristleCount; i++) {
    BristleArrPt [i][0]= BristleArrPt [i][0]* tmp;
    BristleArrPt [i][1]= BristleArrPt [i][1]* tmp;
};

```

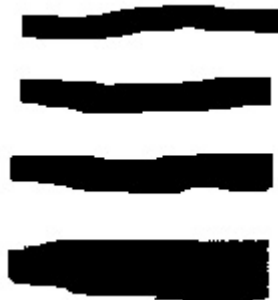
Výpis 9: Přítlak pera

Dalším parametrem, který umí některé modely využívat je snímání rotace pera. Nicméně tablet, který byl využíván pro vývoj aplikace snímat rotaci neumí. Takže vzhledem k tomu, že není možné otestování funkčnosti není rotace pera v aplikaci implementována. Ovšem aplikace je postavena tak, aby bylo možné tuto funkci kdykoliv doplnit. Výpočet štětín při rotaci pera by byl umístěn před výpočtem náklonu. Pro výpočet by byl využit následující vzorec pro rotaci bodu kolem středu.

$$x' = x \cos \alpha - y \sin \alpha$$

$$y' = x \sin \alpha + y \cos \alpha$$

Další funkcí tabletu Wacom, kterou aplikace využívá, je kreslení perem pomocí obou konců. Aplikace automaticky rozpozná, kterým koncem pera se uživatel dotýká plochy tabletu. Díky tomu je druhý konec použit jako guma. Guma byla implementována v aplikaci v podstatě jako druhý štětec se stejným rozmístěním štětín, který na plátno kreslí barvou pozadí. Rozdíl je v tom, že druhý konec pera není ovlivněn náklonem a přitlakem pera.



Obrázek 9: Přítlak pera

Obrázek 10: Nový obrázek

5.3 Kreslicí plátno

Kreslicí plátno je v HTML stránce tvořeno prvkem canvas, který je určen pro vykreslování 2D grafiky. Při spuštění aplikace má uživatel k dispozici kreslicí plátno s aktivní oblastí o rozměrech 1500x1500 pixelů s čistým bílým pozadím. Canvas je na stránce nastavený tak, aby minimální velikost byla vždy stejná jako velikost stránky. Pokud je velikost aktivní plochy menší než rozměry stránky, tak se aktivní oblast vždy nastaví doprostřed canvasu ohraničená barevnou čarou. Barva čáry závisí na barvě pozadí. V opačném případě jsou rozměry aktivní oblasti shodné s rozměry canvasu. Uživatel má vždy možnost vytvoření nového obrázku použitím příslušného tlačítka v nabídce. Kliknutím na tlačítko se uživateli otevře nabídka s možnostmi nového obrázku (viz obrázek 10). Ve vrchní části má k dispozici nastavení velikosti obrázku. Nastavení velikosti je řešeno pomocí textových polí. U tabletů Wacom lze pro jejich vyplnění používat funkci rozpoznávání rukopisu. Velikost obrázku se může pohybovat v rozmezí 32x32 až 2000x2000 pixelů. Zbývající prvky umožňují nastavení barvy a stylu pozadí. Kliknutím na tlačítko *Potvrdit* dojde k vygenerování nového obrázku a zároveň k vyhrazení potřebného místa v paměti pro ukládání informací o obrázku. V paměti je obrázek reprezentován dvěma dvourozměrnými poli, jejichž velikost odpovídá rozměrům obrázku. První pole obsahuje informace o barvě pozadí. Toho je využito při použití gumy, aby nedocházelo ke smazání čar tvořících řádkovaný nebo čtverečkovaný podklad. Opět barva čar stejně jako ohraničení canvasu závisí na zvolené barvě pozadí. Druhé pole pak obsahuje samotné informace o nakresleném obrázku.

Pole s informacemi o pozadí vždy obsahuje informaci v podobě RGB barvy pozadí. Do pole s popředím se však informace o barvě zapisuje jen v případě použití štětce na daném pixelu. Ve

výchozím stavu nebo po použití gumy je pixel reprezentován hodnotou -1. Toto řešení se ukázalo jako nejvýhodnější při přibližování a oddalování obrázku a při ukládání. Při použití nástroje pro přiblížení nebo oddálení obrázku dojde nejdříve k vytvoření nového canvasu o barvě pozadí a k vykreslení čar na pozadí. Barvu pozadí a styl není třeba číst z pole, protože se v průběhu kreslení nemění. Takže postačí vykreslit pozadí a čáry na základě zvolené mezery mezi nimi. Následně dochází k vykreslení obrázku z pole popředí. Program prochází v poli jednu hodnotu za druhou, pokud nalezne informaci o barvě tak v daném místě vykreslí bod. Pokud hodnotu -1, tak ji přeskočí a pokračuje dál. Tento způsob se ukázal mnohem rychlejší než vykreslování bod po bodu z jednoho pole. Nejznatelnější rozdíl byl u velkých obrázků, kdy změna měřítka trvala i několik sekund.

Pokud chce uživatel uložit obrázek do svého počítače, stačí když klikne na tlačítko se symbolem diskety. Program vytvoří v paměti nový prvek canvas, do kterého se uloží všechny informace z aktivní části canvasu, který je viditelný na webové stránce. Pokud by uživatel použil standardní funkci canvasu pro ukládání nebo vygenerování .png souboru, tak by došlo k uložení celé plochy včetně té mimo aktivní oblast. Takže aktivní část canvasu je převedena do nového, který však zůstává uživateli skrytý. Pro vytvoření pozadí je využit stejný postup jako u přibližování a oddalování obrázku. Nejdříve je vykresleno pozadí a poté případně čáry tvořící linkované a čtverečkové pozadí. A stejným způsobem je rovněž vykreslen samotný obrázek. Pro samotné vytvoření .png souboru je využit objekt *Blob* z *FileAPI* knihovny. Blob je rozhraní reprezentující binární data jehož potomkem je objekt *File*, který reprezentuje data ze souborového systému. Před jeho použitím je z dat získaných z canvasu vytvořeno pole celých čísel, s kterým Blob pracuje. Následně je vytvořen nový objekt Blob, který vygeneruje soubor ve formátu .png. Výpis 10 zobrazuje postup při generování souboru pomocí Blob rozhraní. Nejdříve si načteme data z nově vytvořeného canvasu do proměnné *dURL*. Následně si data převedeme do zmíněného pole celých čísel, které pak za pomoci Blob rozhraní převedeme do .png souboru. Bohužel takto nelze u obrázkových souborů přímo vyvolat možnost jejich stáhnutí do počítače. Obrázkové soubory jsou v prohlížeči nastaveny tak, aby se otevřeli v novém okně či záložce. Takže pokud uživatel chce vygenerovaný obrázek uložit, musí tak učinit pomocí místní nabídky vyvolané pravým tlačítkem myši. Posledním krokem je smazání druhého canvasu z paměti.

```
var dURL = newCanvas.toDataURL("image/png");
var data = atob( dURL.substring( "data:image/png;base64,","length" ) ),
asArray = new Uint8Array(data.length);
for( var i = 0, len = data.length; i < len; ++i ) {
    asArray[i] = data.charCodeAt(i);
}
var blob = new Blob( [ asArray.buffer ], {type: "image/png"} );
url = window.URL.createObjectURL(blob);
window.open(url, 'Download');
delete newCanvas;
```

Výpis 10: Vytvoření souboru .png

6 Srovnání

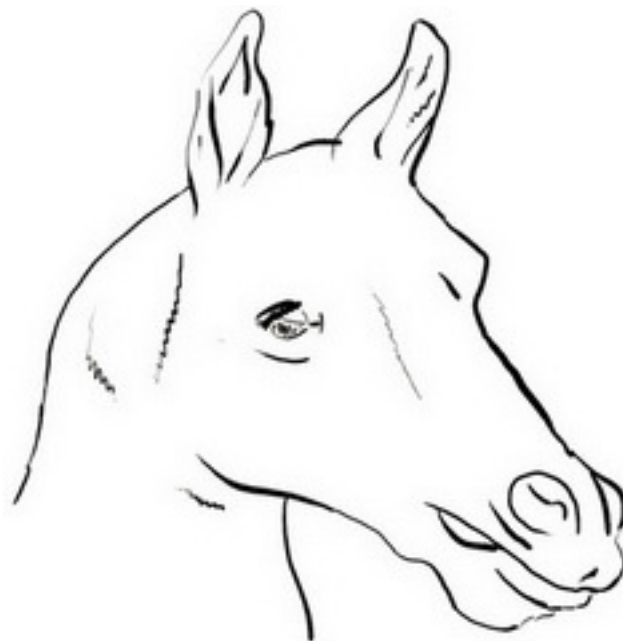
Při vývoji aplikace bylo důsledně dbáno na to, aby aplikace byla pro uživatele co nejjednodušší na ovládání a současně, aby nabízela pokud možno co nejvíce funkcí. Zároveň byla aplikace navrhována, tak aby co nejvíce využila možnosti, které dnes nabízejí jazyky pro tvorbu webových aplikací. Pokud srovnáme vyvíjenou aplikaci s již existujícími programy, které byly popsány ve 3. odstavci, je třeba myslet na to, že tato aplikace je narozdíl od popsaných programů určena pro webovou platformu. Od toho se odvíjeli možnosti, které bylo možno při vývoji využít. Největším problémem bylo, aby aplikace podporovala implementované funkce a neprojevalo se to na běhu aplikace. Díky tomu je možné s aplikací pracovat i na slabších počítačích. Odezvou při kreslení může aplikace konkurovat porovnávaným aplikacím s výjimkou programu Krita Desktop, kde jak již bylo zmíněno dochází na slabších strojích k problémům s během programu a plynulostí kreslení.

Co tato aplikace nabízí je automatická detekce připojeného grafického tabletu a možnost jejich okamžitého použití bez nutnosti cokoli nastavovat. Aplikace umí využívat snímání nejen přítlačku, ale i náklonu pera. Náklon pera ze srovnávaných aplikací umí snímat pouze GIMP. Dále aplikace dokáže automaticky pracovat s oběma konci pera. Pokud uživatel začne kreslit druhým koncem, bude pero automaticky fungovat jako guma. Toto žádná z popisovaných aplikací neuměla. Vždy bylo nutné vybrat příslušný nástroj z panelu nástrojů. Aplikace nadále umožňuje dokreslování detailů při přiblížení obrázku, díky tomu že velikost štětce zůstává bez ohledu na měřítko obrázku stále stejná.

Co však tato aplikace nedokáže, je pracovat s ostrostí štětce a plynulým přechodem. Při kreslení je má stopa štětce, bez ohledu na přítlak, vždy ostré okraje a nepřechází plynule do okolní barvy. Nicméně i přesto lze aplikaci využít pro kreslení jednoduchých kreseb i obrázků. Následující obrázky číslo 11 a 12 ukazují srovnání této webové aplikace s aplikací MyPaint. Z obrázků je vidět, že pomocí webové aplikace, je možné vytvořit obrázky srovnatelné kvalitou s jinými volně dostupnými aplikacemi.



Obrázek 11: Testovací kresba vytvořena ve webové aplikaci



Obrázek 12: Testovací kresba vytvořena v programu MyPaint

7 Testování

Po dobu vývoje programu bylo myšleno na to, aby byl běh programu co nejplynulejší i na méně výkonných sestavách. Proto po celou dobu probíhalo měření odezvy všech důležitých součástí programu. Mezi nejdůležitější část patřilo především měření odezvy při kreslení štětce a při změně měřítka obrázku. První na řadě je kreslení štětce. Při kreslení štětce bylo testováno několik typů štětců s různým počtem štětín, aby se ověřilo jaký vliv má počet štětín na plynulost kreslení. Rovněž bylo ověřeno, jak se na odezvě projevuje výpočet pozic štětín přitlaku a náklonu pera. Tabulka 5 a graf na obrázku 13 zobrazují dobu odezvy v milisekundách. Z tabulky i grafu je patrné, že počet štětín a stejně tak aktivní výpočet parametrů pera se sice v odezvě projevil, ale rozdíl byl minimální. I při zvolení štětce s největším počtem štětín odezva zůstává pod 10 ms.

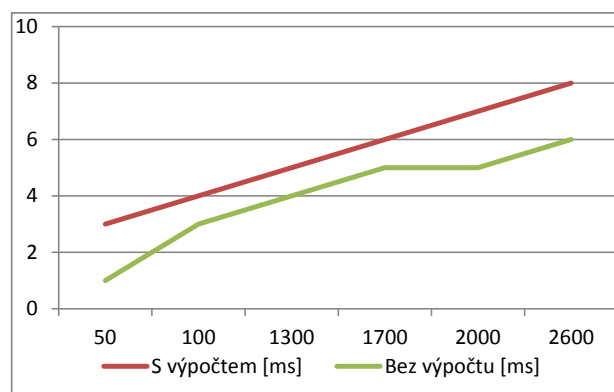
Druhou testovanou částí byla změna měřítka. Byla testována doba od kliknutí na příslušné tlačítko do změny velikosti obrázku. Doba byla především ovlivněna velikostí obrázku a také zvoleným měřítkem. Čím větší rozlišení obrázku, tím větší doba odezvy. S roustoucím přiblížením navíc rostla doba potřebná pro vykreslení obrázku. Rovněž hodnotu odezvy ovlivnil i způsob vykreslování nového obrázku. Jak již bylo zmíněno, tak vrstva popředí používá, pokud nebylo v daném pixelu nic nakresleno, hodnotu -1. Tento způsob zkracuje dobu potřebnou pro nové vykreslení obrázku. V tabulce 6 a grafu na obrázku 14 jsou doby odezvy v milisekundách pro měřítka 100 až 500 %. Jako testovací velikost obrázku byla zvolena 1500x1500 bodů.

Dalšími testovanými hodnotami byla doba odezvy při vytváření nového obrázku a při ukládání. Opět tyto hodnoty byly ovlivněny velikostí obrázku. Čím větší rozlišení obrázku, tím docházelo k prodlužování doby odezvy. Tyto hodnoty jsou obsaženy v tabulce 7 a grafu na obrázku 15. Byly měřeny hodnoty pro nejmenší a největší možné rozlišení. Další hodnoty byly odstupňovány po 500 bodech. Největší dobu u vytváření nového obrázku zabírá vytvoření požadovaného prostoru v paměti počítače a nastavení výchozích hodnot v paměti. U ukládání je to potom převod aktivní části canvasu do nově vytvořeného. Zde však větší odezva není příliš velkou překážkou, protože tyto funkce, narozdíl od předchozích dvou, budou uživatelem využívány jen v případě potřeby.

Mezi měřenými funkcemi byla nakonec i odezva při změně štětce. Zde se však žádné velké rozdíly neobjevovaly. Odezva při změně štětce a následném výpočtu pozic štětín se pohybovala pod 2 milisekundami.

Počet štětín	S výpočtem [ms]	Bez výpočtu [ms]
50	3	1
100	4	3
1300	5	4
1700	6	5
2000	7	5
2600	8	6

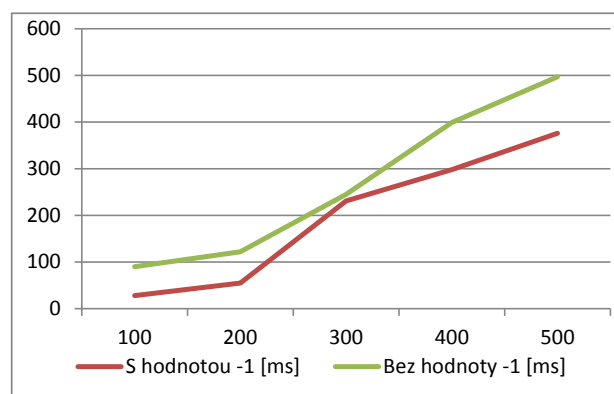
Tabulka 5: Odezva při kreslení štětcem



Obrázek 13: Odezva při kreslení štětcem

Měřítko [%]	S hodnotou -1 [ms]	Bez hodnoty -1 [ms]
100	28	90
200	55	122
300	231	245
400	298	399
500	376	497

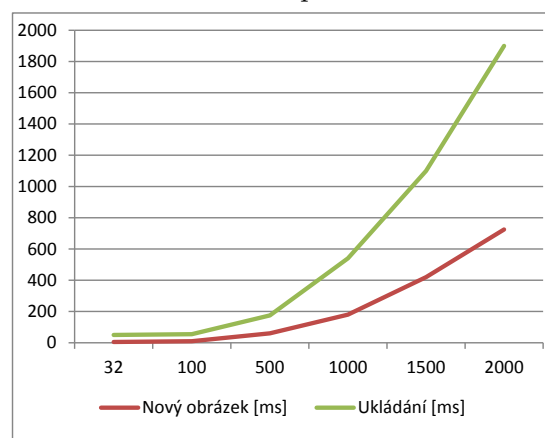
Tabulka 6: Odezva při změně měřítka



Obrázek 14: Odezva při změně měřítka

Velikost [pixel]	Nový obrázek [ms]	Ukládání [ms]
32x32	5	50
100x100	10	55
500x500	60	175
1000x1000	180	540
1500x1500	420	1100
2000x2000	725	1900

Tabulka 7: Odezva při vytváření nového obrázku a při ukládání



Obrázek 15: Odezva při vytváření nového obrázku a při ukládání

8 Závěr

Úkolem této práce bylo vypracovat webovou aplikaci určenou pro grafické tablety Wacom. V prvních dvou kapitolách se práce věnuje samotným grafickým tabletům a existujícím programům, které jsou pro tyto tablety určeny. Jak bylo v těchto kapitolách popsáno, dnešní technologie umožňují uživatelům, vytvářet na počítači profesionální obrázky a umožňují uživateli pracovat na počítači, tak jako by používal skutečný štětec a papír. Tato webová aplikace byla vyvíjena především pro běžné uživatele. Tomu odpovídají i jednotlivé funkce, které aplikace nabízí. Při dalším vývoji aplikace by však bylo možné vylepšit aplikaci, tak aby mohla konkurovat profesionálním kreslicím programům. Hlavní změnou, kterou by však bylo třeba provést je přizpůsobit aplikaci ukončování podpory NPAPI pluginů ve webových prohlížečích.

Mezi další vylepšení, které by do aplikace mohly být implementovány patří například práce s vrstvami nebo paměť úprav. Rovněž by mohl být upraven systém štětců. Současná aplikace neumí pracovat s průhledností a ostrostí štětce. Plynulý přechod mezi barvami by umožnil realističtější kreslení a vzhled obrázku. Rovněž by aplikace mohla být rozšířena o další typy psacích nástrojů, které by napodobovaly jejich skutečné protějšky, tak jako je tomu u řady existujících aplikací. Nicméně už i současná verze se hodí pro základní kreslení v rámci prohlížeče.

Nevýhodou však u této aplikace zůstává, že pokud by ji chtěl uživatel využívat musí být neustále připojený k internetu. V případě kreslicích aplikací by bylo ideálnější vytvářet je jako desktopové. Desktopová aplikace je pro uživatele výhodnější, že ji může využívat kdekoliv a kdykoliv bez nutnosti internetového připojení. Navíc desktopové aplikace jsou použitelné ve všech běžně používaných operačních systémech. Jediné co je třeba pro správnou funkčnost udělat je instalace ovladačů pro tablet. Desktopová aplikace by mohla nabídnout lepší zpracování uživatelského rozhraní a řadu dalších funkcí, které by mohly být v případě webové aplikace problematické. Mezi takovéto funkce by mohlo patřit kreslení pomocí vektorové grafiky. Manipulace s takto nakreslenými objekty by byla programově lépe proveditelná u desktopových aplikací. Rovněž v případě podobných webových aplikací dochází k mnohem většímu zatížení procesoru než je tomu u desktopových aplikací.

Literatura

- [1] BROWN, Paul, et al. *White Heat Cold Logic: Early British Computer Art 1960-1980*. Mit Press, 2009.
- [2] OGAWA, Yasuji. *Information input system by attitude detection of manual implement*. U.S. Patent No 5,561,543, 1996.
- [3] KELBY, Scott; KLOSKOWSKI, Matt. *The Photoshop Elements 8 Book for Digital Photographers*. New Riders, 2009.
- [4] GIMP, G. N. U. Image manipulation program. *User Manual, Edge-Detect Filters, Sobel, The GIMP Documentation Team*, 2008, 8.2: 8.7.
- [5] GRAHAM, Ian S. *The HTML sourcebook*. John Wiley & Sons, Inc., 1995.
- [6] FRAIN, Ben. *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd, 2012.
- [7] FLANAGAN, David. *JavaScript: the definitive guide*. "O'Reilly Media, Inc.", 2006.

Zdroje on-line

- [8] *History of Computer Art - Victoria and Albret Museum* [online]. c2016. Dostupné z: <http://www.vam.ac.uk/content/articles/a/computer-art-history/>
- [9] Wacom Intuos Pro Medium. In: *NZ's Online Computer Store* [online]. c2016. Dostupné z: <http://www.elive.co.nz/>
- [10] Bamboo Img [online]. c2016. Dostupné z: <https://lh4.ggpht.com/>
- [11] *Wacom / Interactive Pen Displays & Tablet Styluses* [online]. c2016. Dostupné z: <http://www.wacom.com/>
- [12] *AMOS Software* [online]. c2016. Dostupné z: <https://eshop.amsoft.cz/>
- [13] TEPLANSKÝ, Ivan. Historie Wacomu. *Adobe noviny* [online]. jaro 2006. Dostupné z: http://www.amsoft.cz/adobenoviny/2006_1/tema/wacom.html
- [14] Wacom Company. *Bamboo Paper* [počítačový program]. Ver. 1.3. c2012. Freeware, dostupný jako součást ovladačů pro tablety Wacom.
- [15] *Intro to Painter Essentials 5 / Corel Discovery Center* [online]. c2016. Dostupné z: <http://learn.corel.com/painting/tutorials/grid/course/Intro-to-Painter-Essentials-5>
- [16] *Krita / Digital Painting. Creative Freedom* [online]. c2016. Dostupné z: <https://krita.org/>

- [17] *My Paint - About* [online]. c2016. Dostupné z: <http://mypaint.org/about/>
- [18] RENOLD, Martin. *MyPaint* [počítačový program]. Ver. 1.2.0. leden 2015. Freeware, dostupný na adrese <http://mypaint.org/downloads/>
- [19] KIMBALL, Spencer; MATTIS, Petter. *GIMP* [počítačový program]. Ver. 2.8.16. listopad 2015. Freeware, dostupný na adrese <https://www.gimp.org/downloads/>
- [20] Wacom Company. *Web Plugins* [online]. 2010. Dostupné z: <http://www.wacomeng.com/web/>
- [21] Wacom WebPlugin FeelTM Multi-Touch API [online]. 2010, poslední revize září 2012. Dostupné z: <http://www.wacomeng.com/web/WebPluginTouchAPI.htm>
- [22] WERKEMA, Sean. *ProColor* [online]. c2009. Dostupné z: <http://procolor.sourceforge.net/>
- [23] *Interactive JavaScript charts for your webpage / Highcharts* [online]. c2016. Dostupné z: <http://www.highcharts.com/>